MCGINN & GIBB, PLLC
A PROFESSIONAL LIMITED LIABILITY COMPANY
PATENTS, TRADEMARKS, COPYRIGHTS, AND INTELLECTUAL PROPERTY LAW
8321 OLD COURTHOUSE ROAD, SUITE 200
VIENNA, VIRGIN IA 22182-3817
TELEPHONE (703) 761-4100
FACSIMILE (703) 761-2375; (703) 761-2376

APPLICATION FOR UNITED STATES LETTERS PATENT

APPLICANT: Koichi Kagawa

FOR: **TABLE MANAGEMENT TECHNIQUE**

DOCKET NO.: **FQ-IP21403**

TABLE MANAGEMENT TECHNIQUE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a table management technique, and in particular to a table management method and device allowing table access using degraded data having the number of bits smaller than that of input data.

2. Description of the Related Art

Hashing is a well-known method that is used to search a table. For example, in a standard LAN (Local-Area Network), MAC (Media Access Control) addresses are uniquely assigned to respective ones of all network devices. In such a LAN system, it is known that the hashing is used to search for a 48-bit MAC address.

Referring to Fig. 1, a 48-bit MAC address is converted to 10-bit data by a hash function 10 and the degraded 10-bit data is used as address data to obtain access to an entry table 11.

Here, it is assumed that the entry table 11 contains 1024 entries, each of which is composed of a MAC address (48 bits), a port number (4 bits) of a switch accommodating such MAC address, an access bit (1 bit) indicating a history of accesses to such MAC address, and a valid bit (1 bit) indicating a

validity/invalidity of registration. Here, the number of bits of a port number depends on the number of ports of the switch. In this case, since a maximum number of 16 ports is assumed, 4 bits are needed for a port number.

Therefore, one entry of the entry table 11 is decided in accordance with a 10-bit address obtained by the hash function, and a registered MAC address included therein is read and output to a comparator 12. The comparator 12 compares the registered MAC address to an input MAC address to judge match/mismatch.

However, since the 48-bit input data is degraded to 10-bit degraded data by the hash function, there is a possibility that different input MAC addresses are mapped into the same address. The frequency of occurrence of this collision depends on how a hash function is selected. In the event of occurrence of a collision, rehashing is performed to change the hash function so as to generate different hash outputs for different input MAC addresses.

The rehashing causes all of contents stored in the entry table 11 to be invalid, resulting in significant performance reduction of MAC address learning. Therefore, in consideration of the hashing mechanism, it is an important issue how the occurrence of rehashing is suppressed.

SUMMARY OF THE INVENTION

An object of the present invention is to provide a table management method and device allowing efficient hash search with suppressing the possibility of occurrence of rehashing.

According to the present invention, a table management device uses degraded data as address data, wherein the degraded data is produced from input data having a predetermined number of bits, the degraded data having a smaller number of bits. The table management device includes: a plurality of tables allowed to be concurrently accessed according to the degraded data, wherein each of the tables is allowed to register a predetermined number of pieces of data, each of the pieces of data having a number of bits equal to the predetermined number of bits of the input data; a plurality of comparators provided for respective ones of the plurality of tables, wherein each of the comparators compares the input data to a piece of data read from a corresponding table according to the degraded data to produce a comparison result; and a determiner for determining from comparison results of the comparators whether the input data has been already registered in the tables.

The table management device may further include: a controller controlling the tables such that, when the input data has not been registered in the tables and an available memory area exists in memory space of the tables concurrently accessed according to the degraded data, the input data is registered as new data in the available memory area of the tables.

Each of the comparators may compare a corresponding piece of data to the input data to produce a comparison result indicating one of match and mismatch. The determiner may determine that the input data has been registered in the tables when a match-indicating comparison result is received from at least one of the comparators, and determines that the input data is not registered in the tables when a mismatch-indicating comparison result is received from each of the comparators.

According to another aspect of the present invention, a method for managing an address table which is divided into a plurality of banks, includes the steps of: a) converting input address data having a predetermined number of bits to address data having a smaller number of bits according to hash processing; b) concurrently accessing the banks according to the address data to read registered address data from each of the banks; c) comparing the input address data to the registered address data read from each of the banks to produce comparison results for respective ones of the banks; and d) determining from the comparison results whether the input address data has been registered in the address table.

The method may further include the steps of: e) when the input address data has not been registered in the address table, determining whether an available memory area exists in memory space of the banks concurrently accessed according to the address data; f) when an available memory area exists, registering the input address data as new address data in the

available memory area; and g) when no available memory area exists, changing the hash processing.

As described above, according to the present invention, a plurality of tables are simultaneously accessed by the degraded data. The respective registered data having been read out are compared to the input data and the comparison results are used to determine whether the input data has been registered or not. Since a plurality of registered data is simultaneously accessed to read out, an extremely effective search can be performed.

Further, when the input data has not been registered at the tables and there is a free memory area in memory space of the tables accessed simultaneously by the degraded data, the input data is registered as new data in the available memory area. Therefore, a plurality of different input data can be stored in the memory space of the tables accessed simultaneously by the same degraded data. In other words, since one degraded data can be associated with a plurality of different registered data, the probability of occurrence of rehashing can be significantly reduced. That is, only when there is left no memory area in the accessed memory space of the banks, a hash function or an algorithm for generating the degraded data from the input data is changed. Thus, compared with the prior art, the probability of occurrence of rehashing is significantly reduced.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing a conventional management device which manages a MAC address table;

Fig. 2 is a block diagram showing an address table management device according to an embodiment of the present invention;

Fig. 3 is a flow chart showing a registration/learning operation of a MAC address table according to the present embodiment; and

Fig. 4 is a schematic diagram showing an entry table for explaining a new address registration operation in the present embodiment.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to Fig. 2, an address table management device according to an embodiment of the present invention is provided with an entry table 102 which is divided into a plurality of banks. For simplicity, assuming that the entry table 102 is divided into four banks B1 - B4, each of the banks B1 - B4 stores

a maximum number of 256 entries and therefore the entry table 102 can store a total of 1024 entries (= 254 entries × 4 banks). As the conventional case, each entry is composed of a 48-bit MAC address, a 4-bit port number, a 1-bit access bit, and a 1-bit valid bit (see Fig. 4).

The hash function 101 performs the CRC 32 calculation of an input 48-bit MAC address and selects eight (8) bits at a predetermined position out of 32-bit output obtained by the CRC 32 calculation to output the selected 8-bit data to the entry table 102. The banks B1 - B4 of the entry table 102 simultaneously input the selected 8-bit data as address data from the hash function 101. Therefore, a single hash output (8-bit address data) allows simultaneous access to memory areas of the banks B1 - B4 specified by the hash output.

When at lest one entry exists in the accessed memory areas of the banks B1 - B4, the registered MAC address included in each entry is simultaneously read out. Since the maximum number of entries in each bank is 256, 8 bits are needed to discriminate the entries. For example, in the case where a table having 1024 entries is equally divided into 8 banks with the maximum number of 128 entries for each bank, the output of hash function 101 is set to 7 bits.

Comparators C1 - C4 compare four registered MAC addresses read out from the banks B1 to B4 to the input MAC address, respectively. The respective comparators C1 - C4 output comparison results (match or mismatch) to an OR circuit 103.

When the OR circuit 103 detects a match in at least one of comparison results, the OR circuit 103 outputs a match detection signal to a processor 104. If no match is detected, the OR circuit 103 outputs a mismatch detection signal to the processor 104.

The processor 104 is a program-controlled processor such as CPU or a dedicated hardware circuit, which is designed to manage the entry table 102. The processor 104 executes a table management program to perform registration (learning) and searching processes for the banks B1 - B4 while monitoring the detection result (match or mismatch) given by the OR circuit 103, which will be described hereafter. The table management program is stored in a recording medium (not shown) such as a read-only memory, a magnetic disk, an optical disk, or a magneto-optic disk.

Registration or Learning Process

It is assumed that some MAC addresses have been already registered in each bank of the entry table 102. A registration procedure for registering a new MAC address in the banks B1 - B4 will be described with reference to Figs. 3 and 4.

Referring to Fig. 3, when the source MAC address of a packet is inputted (step S201), an 8-bit address is calculated as a degraded address by the above-mentioned hash function 101 (step S202). The respective memory areas of the banks B1 - B4 specified by the 8-bit address are simultaneously accessed to read corresponding entries, and when a MAC address is validly

registered in the corresponding entry, it is read out (step S203).

When receiving the registered MAC addresses from the banks B1 - B4, the respective comparators C1 - C4 compare the registered MAC addresses to the source MAC address and output comparison results (match or mismatch) to the OR circuit 103. As described above, the OR circuit 103 outputs a match detection signal to the processor 104 when at least one of the comparison results indicates a match. Contrarily, when all the comparison results indicate a mismatch, the OR circuit 103 outputs a mismatch detection signal to the processor 104.

When receiving a match detection signal from the OR circuit 103 (YES in step S204), the processor 104 determines that the input source MAC address has been already registered in the entry table 102 and therefore does not perform a registration/learning process.

When receiving a mismatch detection signal from the OR circuit 103 (NO in step S204), the processor 104 determines that the input source MAC address is a new address and then determines whether the four accessed memory areas of the banks B1 - B4 include free space (step S205).

When available memory space exists in the four accessed memory areas (YES in step S205), the source MAC address is registered as a new address at an available one of the four accessed memory areas (step S206). This is a learning process. When all of the four memory areas have been occupied (NO in step

S205), the rehashing operation is executed (step S207). As an example, the rehashing is carried out by clearing all of valid bits and selecting 8-bit data at different position from the 32-bit CRC 32 output.

As shown in Fig. 4, it is assumed that memory space 301 of the banks B1 - B4 in the entry table 102 is accessed by an 8-bit address which is an output of the hash function 101. It is further assumed that respective areas of the banks B2 and B4 have already registered MAC addresses Aa and Ab, and respective areas 302 and 303 of the banks B1 and B3 are available.

Even if a source MAC address Ac produces the same hash output indicating the memory space 301, as long as the source MAC address Ac is different from either of the registered MAC addresses Aa and Ab (mismatch : NO in step S204 of Fig. 3), this source MAC address Ac is registered as a new address in, for example, a free area 302 of the bank B1. Similarly, even if another source MAC address Ad produces the same hash output indicating the memory space 301, if it is different from either of the registered addresses Aa, Ab and Ac, then the source MAC address Ad is registered as a new address in a free area 303 of the bank B3. In this way, four different MAC addresses can be registered for the same hash output in the entry table 102.

The rehashing occurs only when another source MAC address Ae produces the same hash output indicating the memory space 301 having no memory space left. In the present embodiment, since up to four MAC addresses can reliably be registered, the

frequency of occurrence of rehashing can be significantly reduced.

Also, the present embodiment allows high-speed learning because the four registered MAC addresses are simultaneously read out and are simultaneously compared to the source MAC addresses by the four comparators C1 - C4.

Searching Process

When a destination MAC address of a packet to be forwarded is input, an 8-bit address is calculated by the above-described hash function 101. The memory areas of banks B1 - B4 at the 8-bit address are simultaneously accessed, and when a valid MAC address is registered in the accessed memory areas, it is read out. Then, the respective comparators C1 - C4 compare the read-out registered MAC addresses to the destination MAC address, and respective comparison results (match or mismatch) are output to the OR circuit 103. As described above, the OR circuit 103 detects match or mismatch depending on whether at least one of comparison results indicates match or all comparison results indicate a mismatch.

When a match is detected, the processor 104 reads out the port number of a corresponding registered MAC address and designates it as a forwarding destination of the packet. When a mismatch is detected, the processor 104 forwards the packet as a broadcast packet to all output ports of the switch.

Also, in the present embodiment, although the entry table 102 is divided into 4 parts, it may be divided into an

arbitrary number N of parts. In the case where N is larger than 4, since N different addresses can be registered for the same hash function value, the probability of occurrence of rehashing is further decreased.

In the case of the entry table divided into a plurality of banks, there are needed as many comparators as the banks. However a comparator has a simple circuit, it does not become a significant burden on a system.

Architecture of the present invention can be applied to not only the management of a MAC address table but the table searching or data registration techniques. Further, arbitrary hash function may be employed.